Controllability

"Chemical makers are not required to test their compounds for toxicity
before asking the E.P.A. (Environmental Protection Agency) to clear them for sale,
and the agency said that most companies did not test voluntarily.
The E.P.A. relies instead on methods, like computer modeling,
to prevent harmful compounds from entering the market."
—quoted from "Will Environmental Fear Stick to DuPont's Teflon?"
New York Times, 24 July 2005

I'd be thrilled if some algorithm or technique I developed were used on a real process. But I'd also worry about this kind of scenario:

Prosecutor: Were you aware of the possibility that biased estimates could destabilize the boiler?
EIC: Yes, bias can occur when signals have correlated noise. That's standard knowledge in control system engineering.
Prosecutor: And can you please tell the court what you did to prevent that from happening?
EIC: I ran a lot of simulations to verify reliability.
Prosecutor: And you're sure you ran *enough* simulations? (skeptical look)

When you think about it, control isn't for the faint of heart. We develop hardware and software so computers can run things with minimal human intervention. We analyze, simulate, and test—over and over—until we satisfy some statistical specification, or until we run out of time or money. And then we ask people to trust that what we built is safe and reliable.

Speaking of safety, the "Applications of Control" column in the October 2005 issue of *IEEE Control Systems Magazine* discussed a possible future car whose onboard processor might prevent sudden turns that can destabilize a vehicle. This feature sounds great unless I need to avoid hitting someone running into the street. Liability, you say? Looks like busy times for JDs.

Heights of distinction. The top of Engineer Mountain was the site of Engineer City, once a silver mining town and now abandoned.

Water and smoke. The Durango-Silverton steam-powered trains have carried passengers past breathtaking Colorado scenery since 1881. The meticulously maintained engine and cars are seen here leaving Durango.

Reliability is always an issue in control system technology. In a hotel where I recently stayed, the elevator button sometimes got stuck. Late at night when no one was around except curious control engineers, the "automatic" door would open, wait a few seconds, and then close. At that moment, the elevator "realized" that the button was pressed (in fact, the button was never unpressed) and, guess what? The door would open again. This went on all night long.

While observing this marvel of control technology, my thought was "Why didn't the elevator control gurus insert a line of code that says 'if the door opens and shuts three times within 20 seconds, then call someone to report a stuck button' or something like that?" Without that line of code, the elevator looked just plain dumb. (By the way, I reported this problem to the maintenance people, mentioning in passing that I'm an engineer. I prudently omitted what *kind* of engineer.)

I notice that a lot of researchers are working on fault detection, which seems like a hard problem. In fault detection, the control system must also determine whether the hardware has failed. That's not an easy task. Have you ever been driving and thought you might have a tire problem, only to find out that the road was rough? It's tricky to distinguish hardware failures from anomalous measurements. Having to doubt the sensors makes things complicated.

I've heard complaints that users of control technology are slow to adopt innovations. If there isn't much to gain and there's a lot to lose by trying something new, then I can see why. Maybe we're victims of our own criticality. Or maybe our present tools, which never seem to be quite good enough, are just a little too good. Yet we also rarely talk about the systems we couldn't build because our tools and techniques weren't good enough. In practice, we're limited to building systems that we can control and understand.

Control technology has some pretty serious implications. We ask people to place their lives in our loops, which makes us responsible for how reliably these systems work. Although control technology is often invisible, the consequences of failure are not. Something to think about.

Dennis S. Bernstein
Editor-in-Chief
IEEE Control Systems Magazine